



**Jet Propulsion Laboratory**  
California Institute of Technology

## F' (F Prime)

A Small Scale Reusable Component Framework for Space

Aadil Rizvi

Brian Campuzano

4/19/2017

© 2017 California Institute of Technology.  
Government sponsorship acknowledged.



# Special Thanks

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

Original authors of materials:

Tim Canham – F' Architect and lead of Mars Helicopter  
Technology Development

Garth Watney – Modeling Advisor and lead of F' modeling  
task



What software characteristics would you favor in a Space Software architecture?



# F' Goals

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

Goal	Explanation
Reusability	Frameworks and adaptations readily reusable
Modularity	Decoupled and easy to reassemble
Testability	Components easily isolated for testing
Adaptability	Should be adaptable to new contexts and bridge to inherited
Portability	Should be portable to new architectures and platforms
Usability	Should be easily understood and used by customers
Configurability	Facilities in the architecture should be scalable and configurable
Performance	Architecture should perform well in resource constrained contexts. Should be very compact.





# Terminology

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY



- Commands
- Telemetry
- Events
- Parameters
- Faults
- Sequences
- Health



# What is F'?

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- F' (F Prime)
  - Targeted for instruments, CubeSats and other smaller platforms
  - Currently baselined for JPL Leon3 based CubeSat avionics processor
- A component-based architecture as well as a software and testing framework to support it
- Designed from the ground up to be compact and reusable
- Includes framework, code generators, build tools, Command/Telemetry GUI, and test environment
- Designed to make it easier for developers to concentrate on mission-specific logic rather than common implementation patterns.



# Where is it being used?

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Development
  - Developed under software/hardware technology tasks (2013-2016)
  - Using established flight software practices and tool checking
- Flew on RapidScat (2014-2016)
  - Radar experiment on International Space Station
  - No reported software bugs
- Baselined for:
  - Leonardo (Mars Helicopter Technology Development)
  - Asteria (Cubesat)
  - Lunar Flashlight (Cubesat)
  - NEAScout (Cubesat)
- Available to anyone at JPL, soon to be open sourced
  - Reference example can be run on Linux, MacOS, Cygwin, Raspberry Pi and JPL embedded flight hardware platforms.

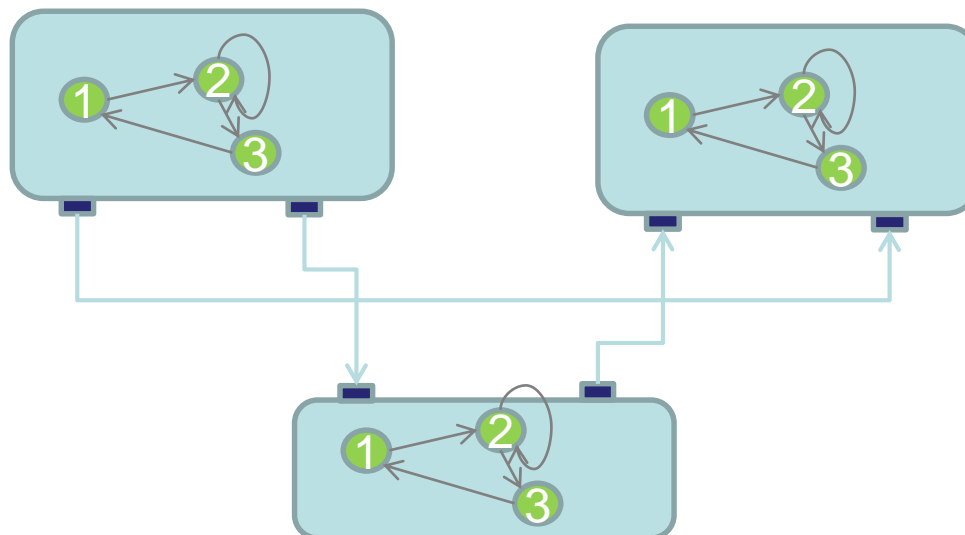


# F': A Reusable Component Architecture

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Consists of components (behaviors) and ports (interconnections for data)
- Components are not dependent on other components, so can be reused.
- Components to fulfill different requirements (simulation vs. actual) can be substituted, even at run time.
- Components can have generic roles (commanding, telemetry, storage) which are not dependent on specific applications.





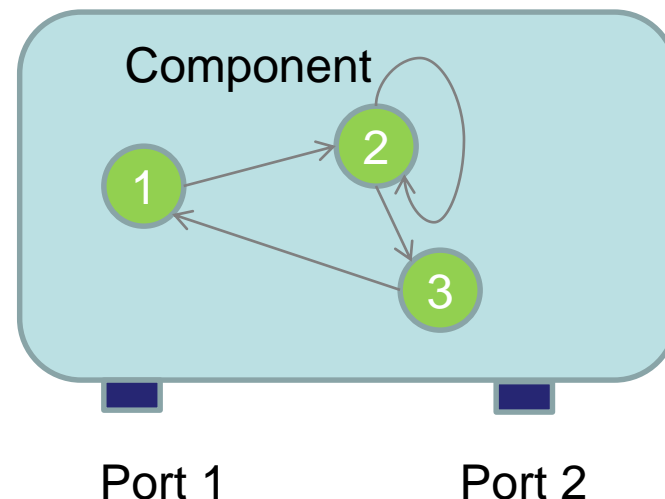


# F': A Framework for quick development

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- F' provides a C++ framework and code generator that encapsulates:
  - Thread management
  - Inter-Process communication (IPC)
  - Commanding
  - Telemetry
  - Parameters
- Since these are common patterns, developer specifies in simple XML.
  - Code generator generates boiler-plate code.
  - Developer concentrates on domain-specific code.
  - Framework invokes user code automatically



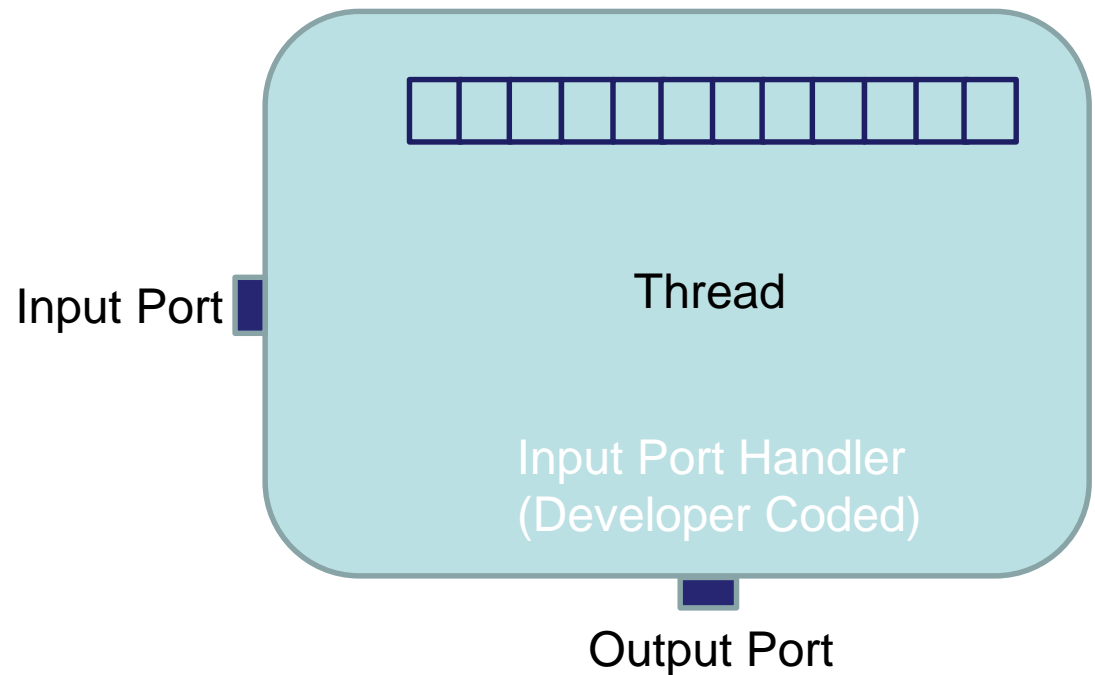


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- **Active**
- Queued
- Passive



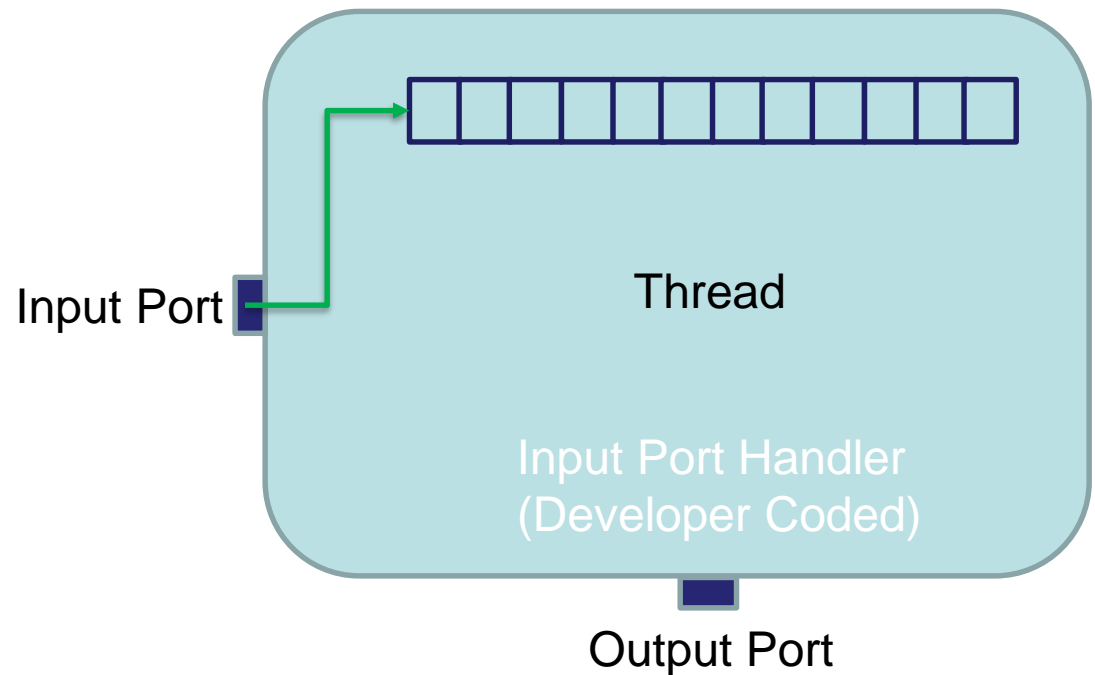


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- **Active**
- Queued
- Passive



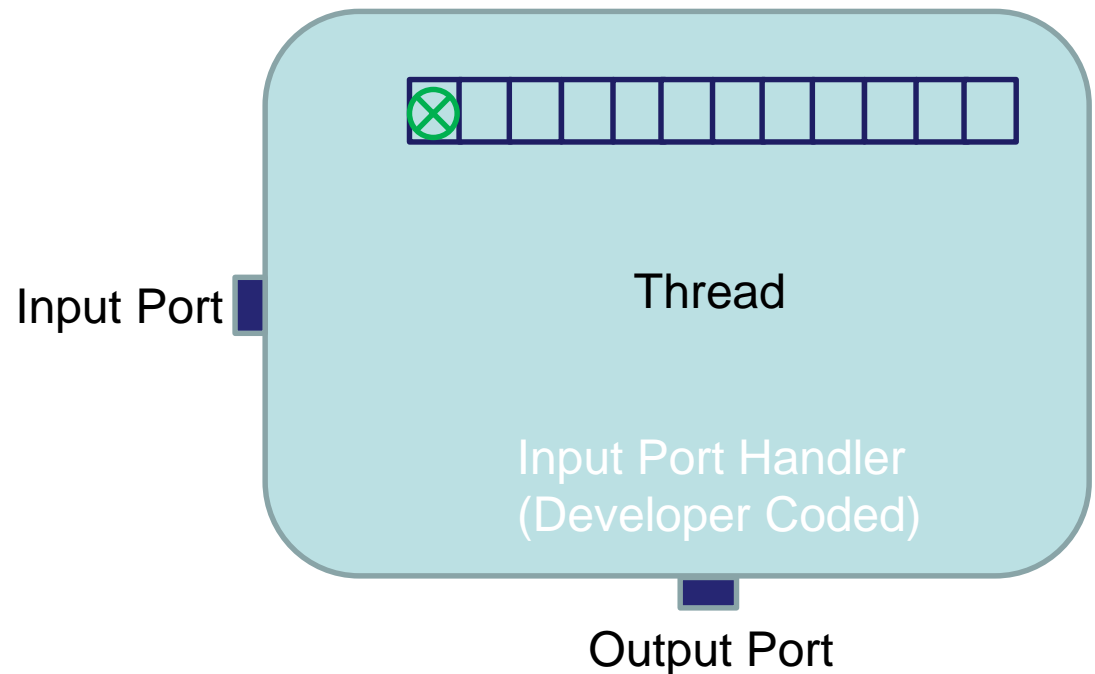


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- **Active**
- Queued
- Passive



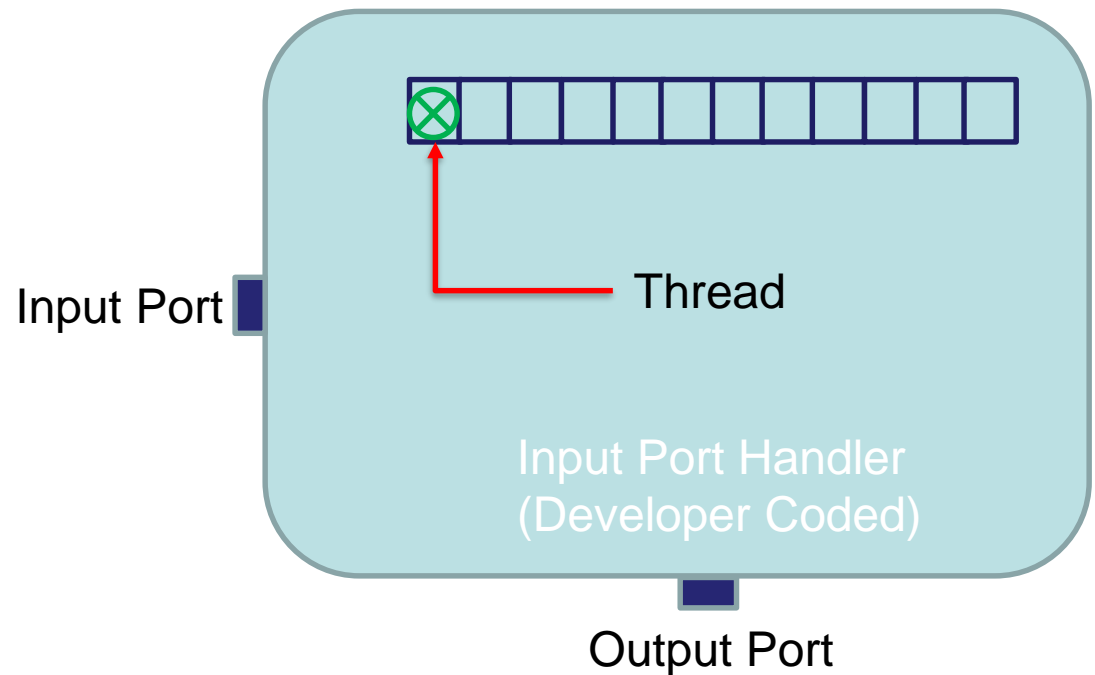


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- **Active**
- Queued
- Passive





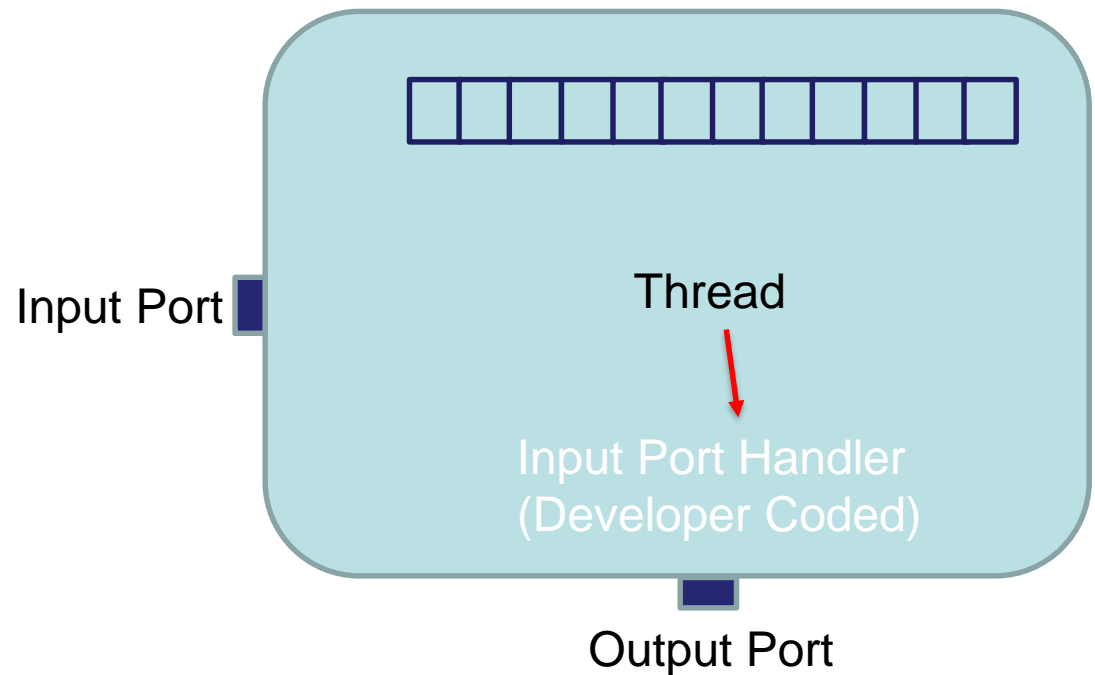


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- **Active**
- Queued
- Passive



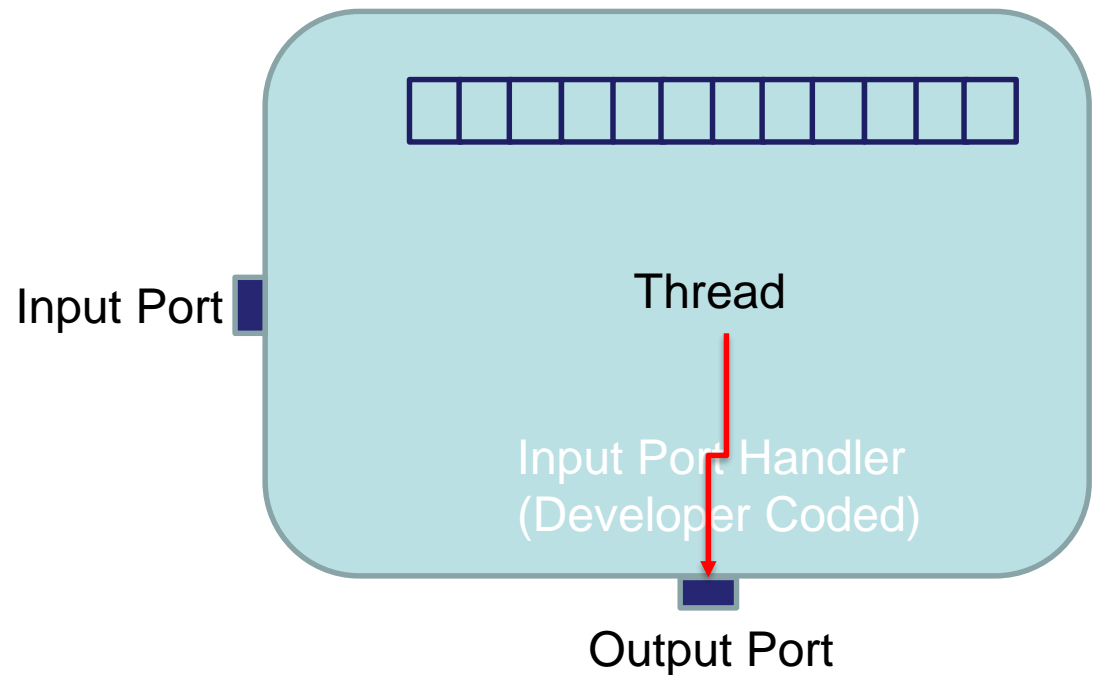


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- **Active**
- Queued
- Passive



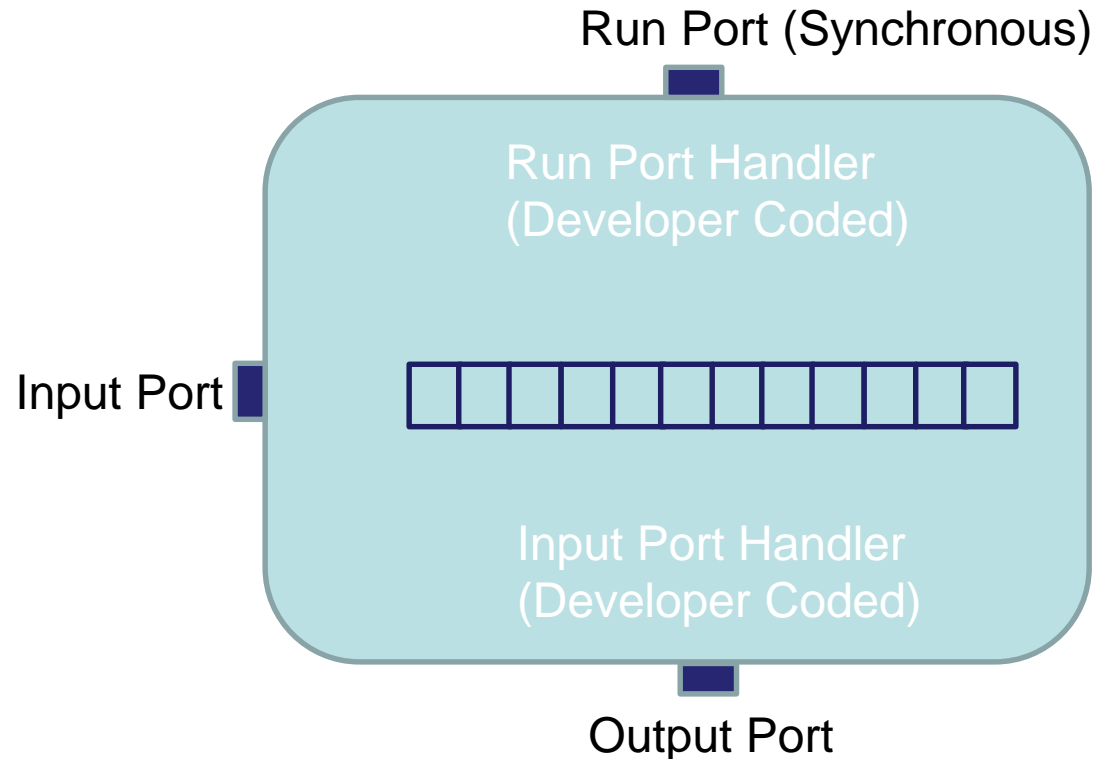


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Active
- **Queued**
- Passive



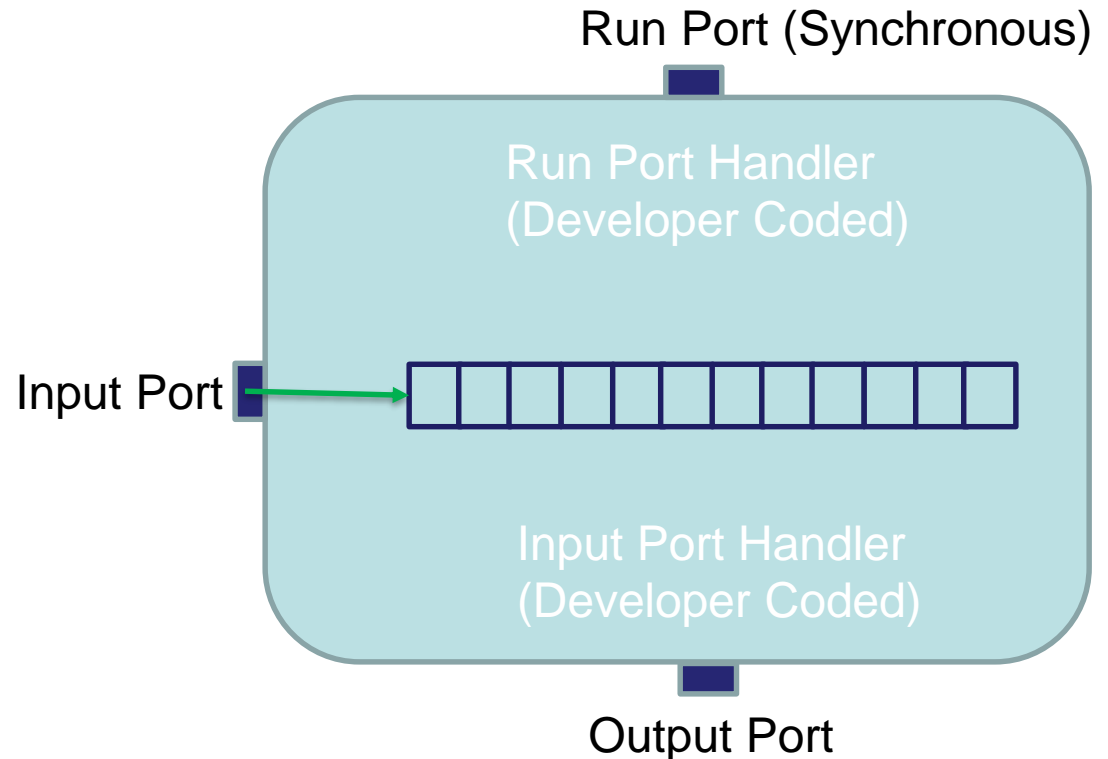


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Active
- **Queued**
- Passive



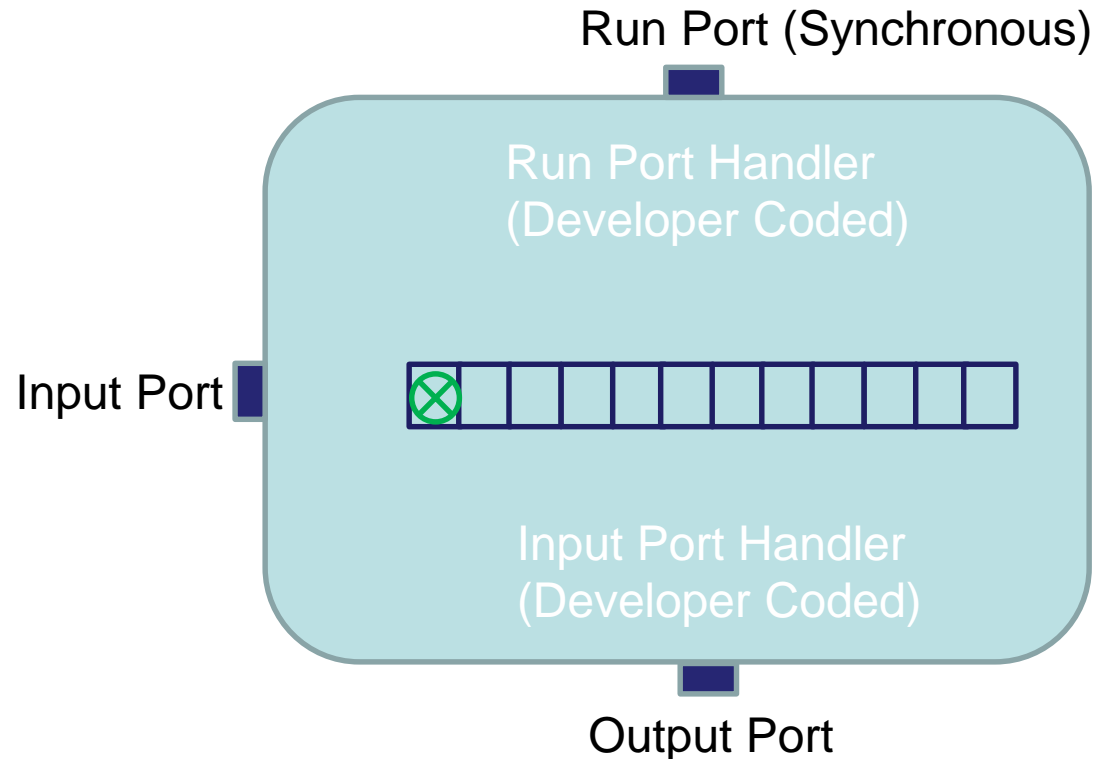


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Active
- **Queued**
- Passive





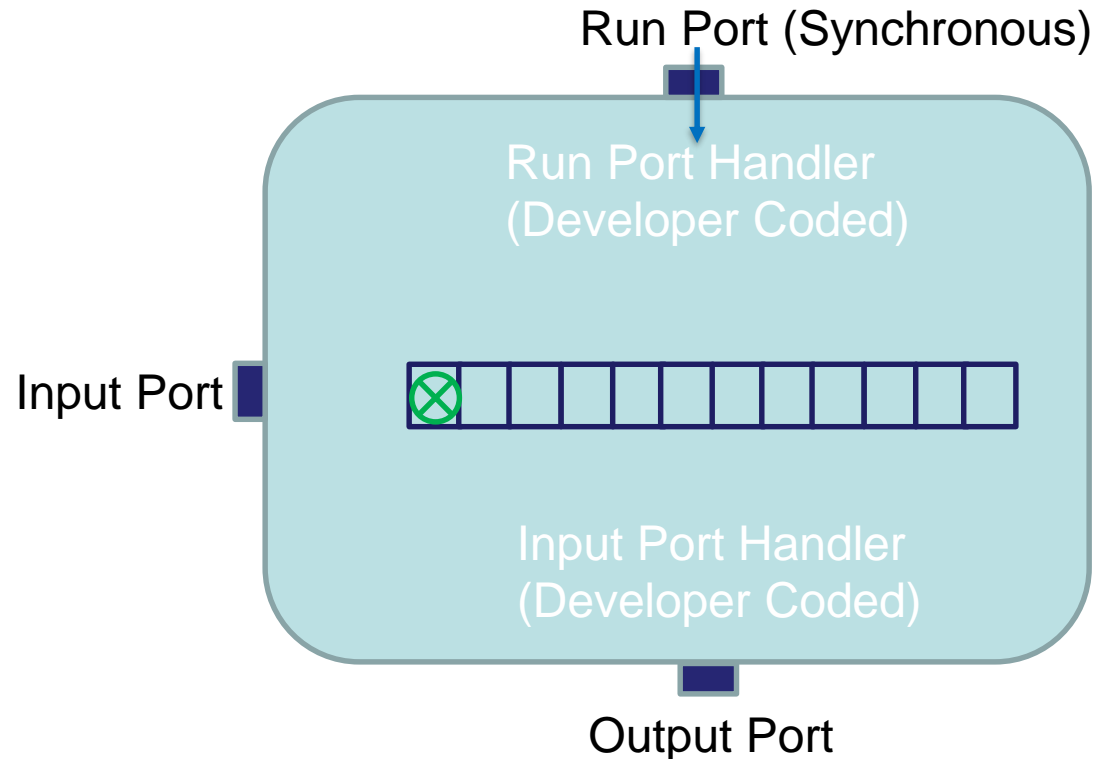


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Active
- **Queued**
- Passive



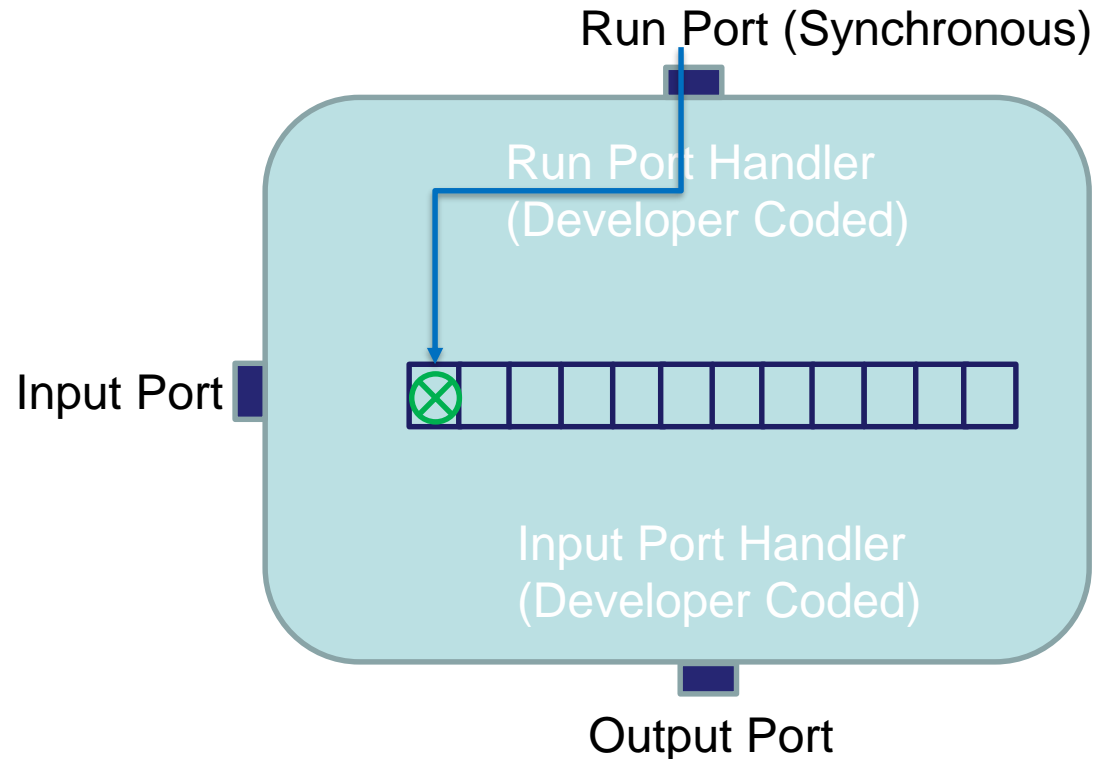


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Active
- **Queued**
- Passive



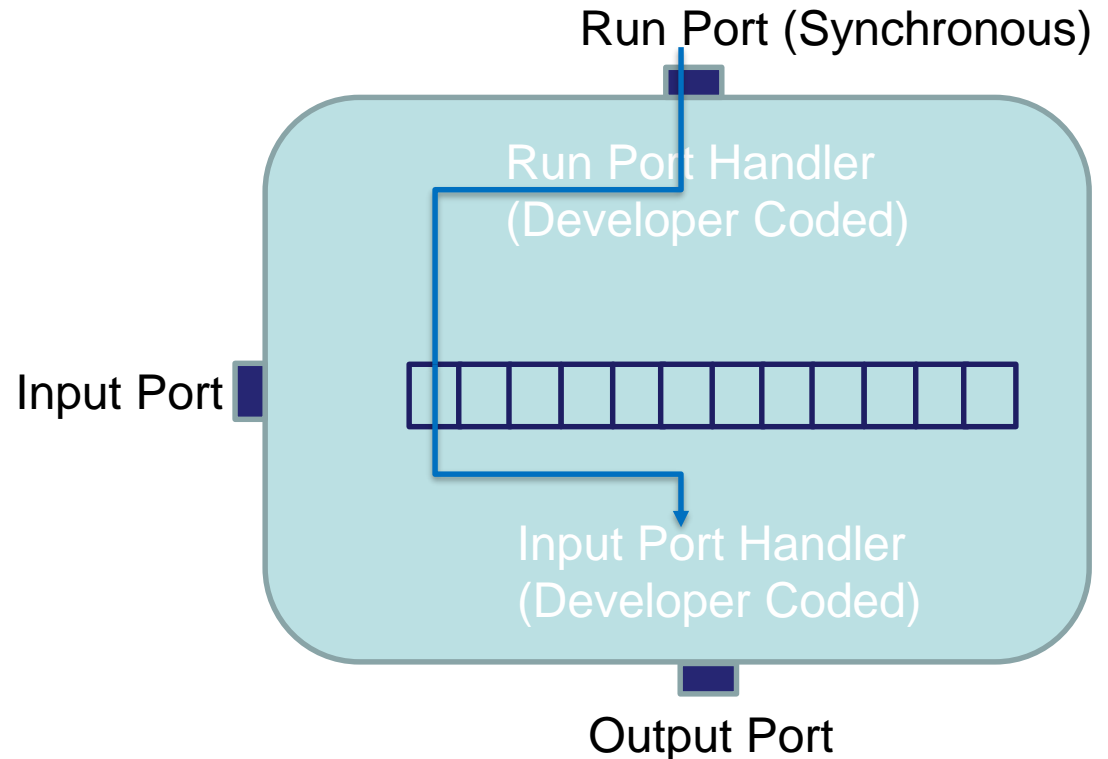


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Active
- **Queued**
- Passive



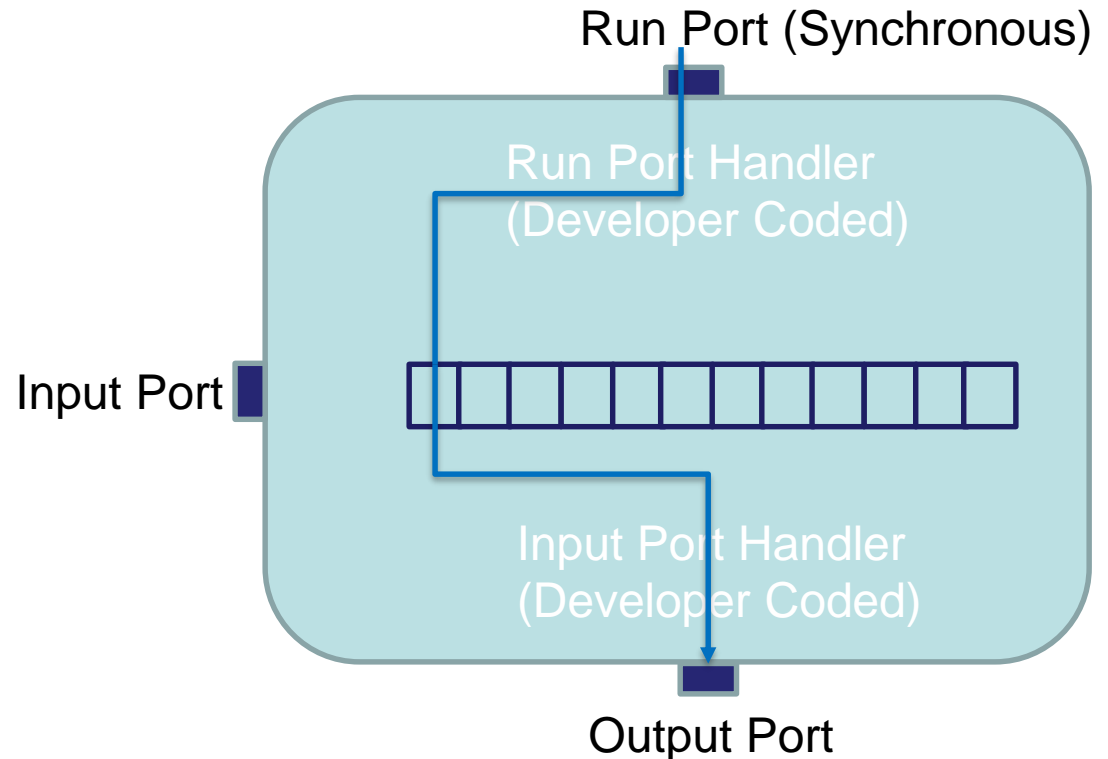


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Active
- **Queued**
- Passive



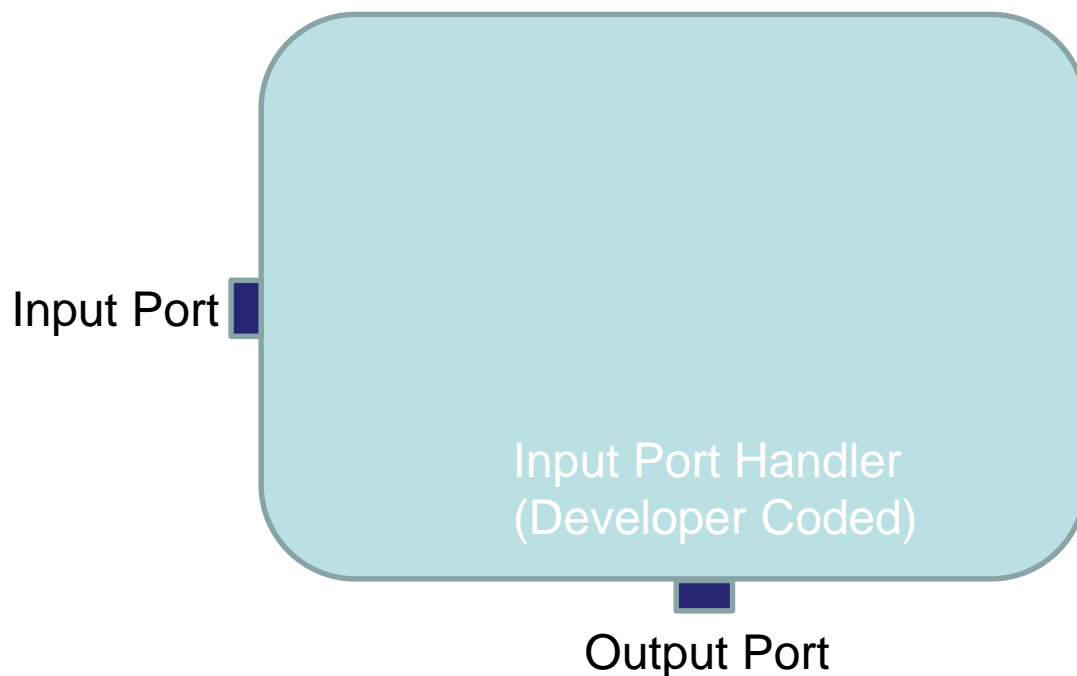


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Active
- Queued
- **Passive**





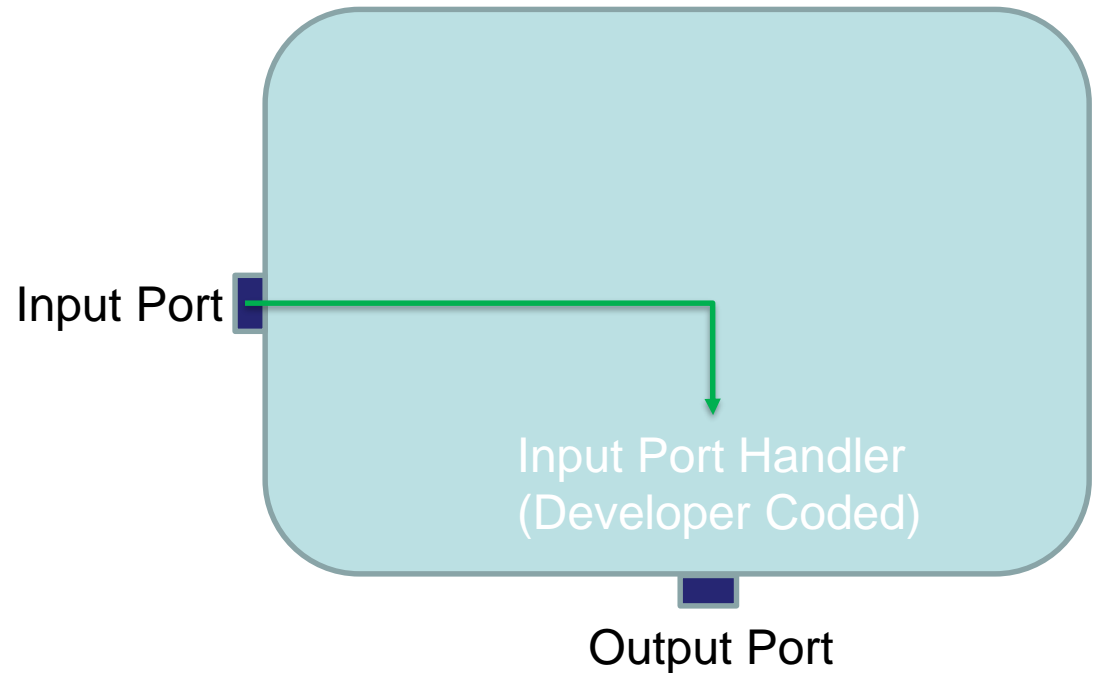


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Active
- Queued
- **Passive**



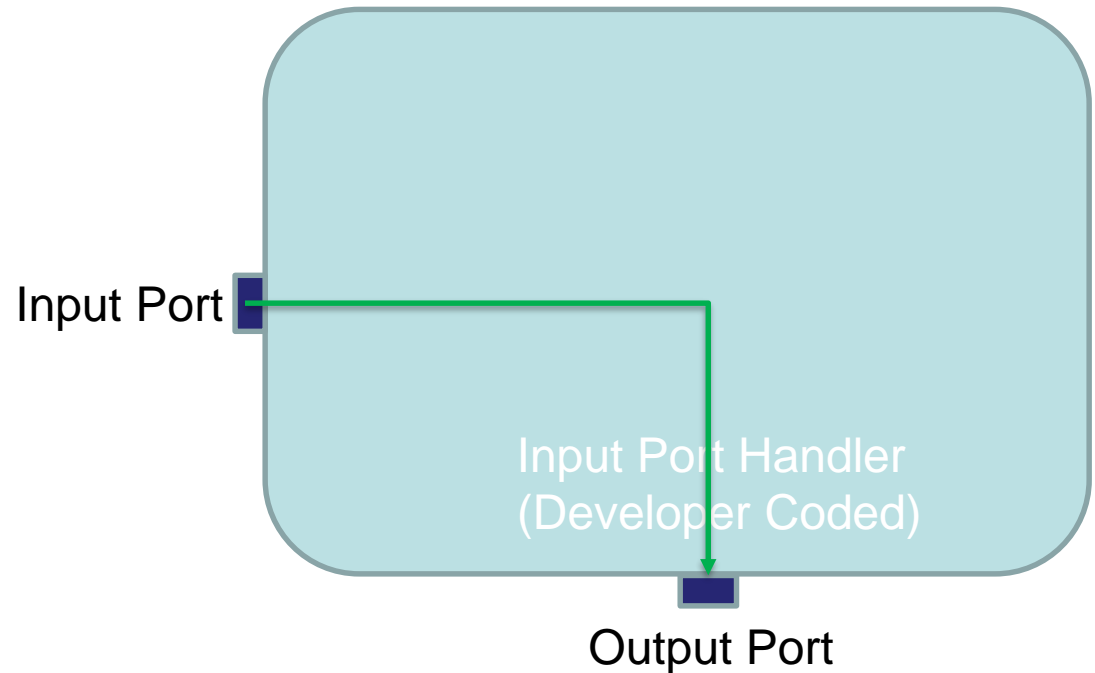


# F': Component Types

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Active
- Queued
- **Passive**



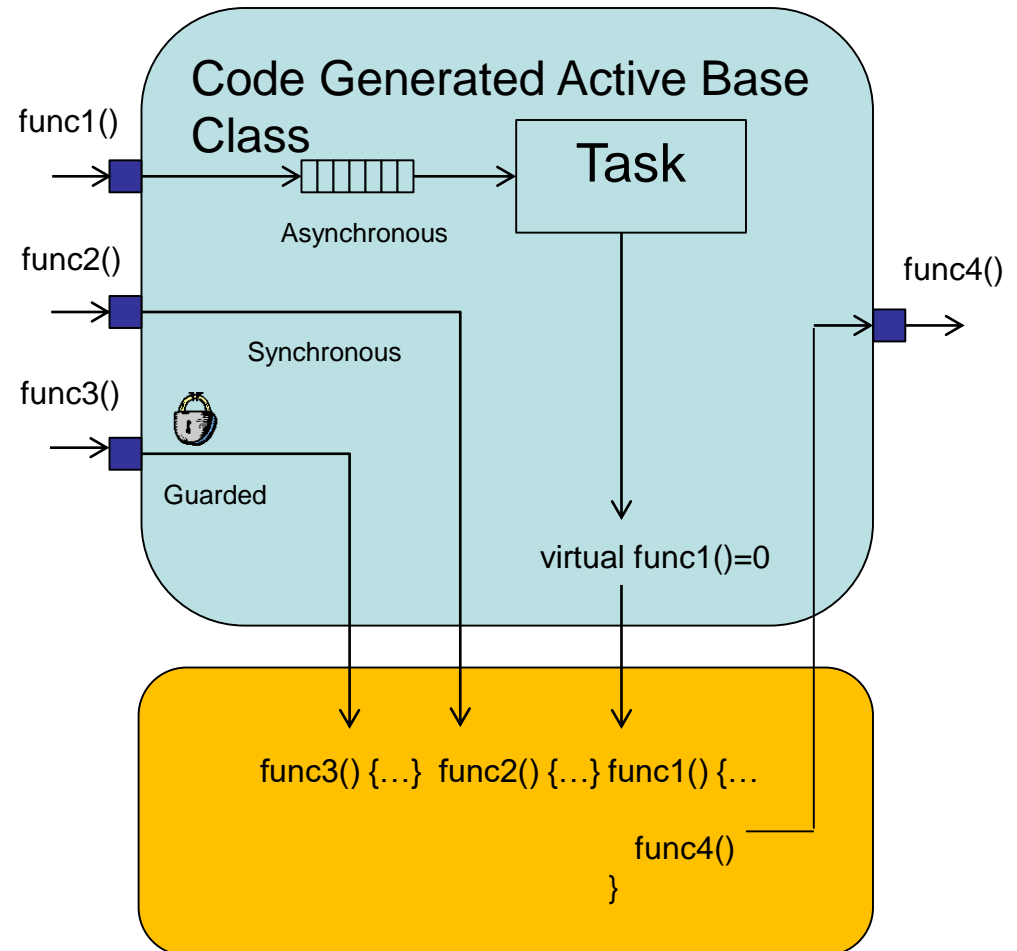


# Port Characteristics

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- The way incoming port calls are handled is specified by the component XML.
- Input ports can have three attributes:
  - Synchronous – port calls directly invoke derived functions without passing through queue
  - Guarded – port calls directly invoke derived functions, but only after locking a mutex shared by all guarded ports in component
  - Asynchronous – port calls are placed in a queue and dispatched on thread emptying the queue.
- A passive component can have synchronous and guarded ports, but no asynchronous ports since there is no queue. Calls execute on the thread of the calling component.
- A queued component can have all three port types, but it needs at least one synchronous or guarded port to unload the queue and at least one asynchronous port for the queue to make sense.
- An active component can have all three varieties, but needs at least one asynchronous port for the queue and thread to make sense.
- Designer needs to be aware of how all the different call kinds interact (e.g. reentrancy)
- Output ports are invoked by calling generated base class functions from the implementation class.



Developer Written Implementation Class



# F': A Portable Framework

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- Code base is in portable, embedded C++
- Has abstraction layer for OS facilities such as:
  - Threads
  - Synchronization
  - Files
  - Time
- Data is stored and transmitted in a portable representation
  - Allows interaction with ground system no matter the processor architecture
- Has been run on the following processor architectures:
  - X86, PPC, ARM, MSP430, Leon3
- Has been run on the following OSes:
  - VxWorks, RTEMS, Linux, MacOS, Cygwin, Raspberry Pi Raspbian
  - Run on microcontrollers with no OS
- Very compact
  - Framework classes ~40K compiled



# F': Layered Architecture

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

Utilities

Apps

Services

Drivers

OS



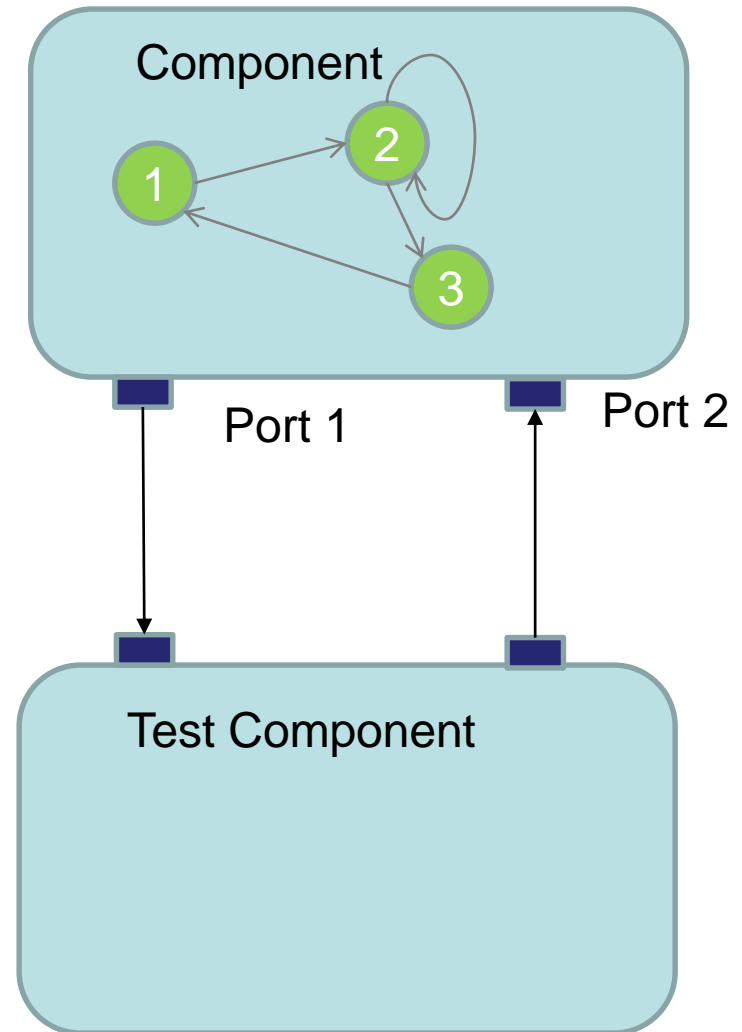


# F': A Framework for testing

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

- F' components are decoupled from others, so unit testing is easier
- F' code generator generates counterpart test component that can be connected.
- Test component “knows” the interfaces, commands, and telemetry
- Tester can invoke generated C++ functions to exercise component interfaces, commands.
- Telemetry automatically decoded and stored for checking in test component.





# F': A Flight-ready Framework

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

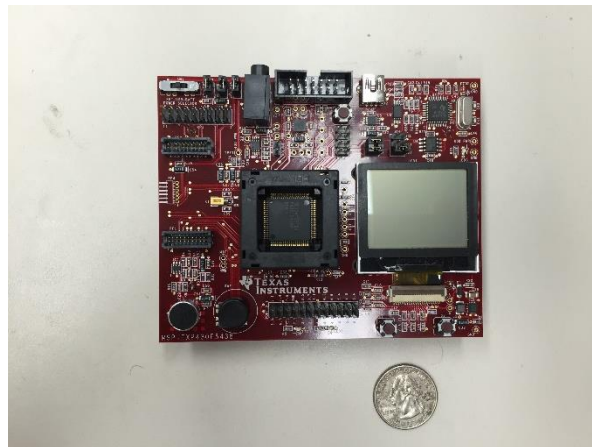
- C&DH components have been taken through flight software practices
  - Design, coding and testing reviews with static analyzer tools and code coverage
    - Design and code reviewed by peers
    - Code scrubbed by JPL institutional analyzer tools
    - 100% coverage except certain assertions (default switch, etc)
    - Delivered with repeatable automated unit tests
  - Includes:
    - Rate Groups
    - Command handling
    - Telemetry Processing
    - Parameter storage
    - Event handling
    - Sequencing
    - File Uplink/Downlink
    - Health Monitoring
    - Fault Protection
    - Thermal Control



# Both Ends of the Scale

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY



TI MSP430

- 24K RAM
- 64K Flash



Rack Mount PC

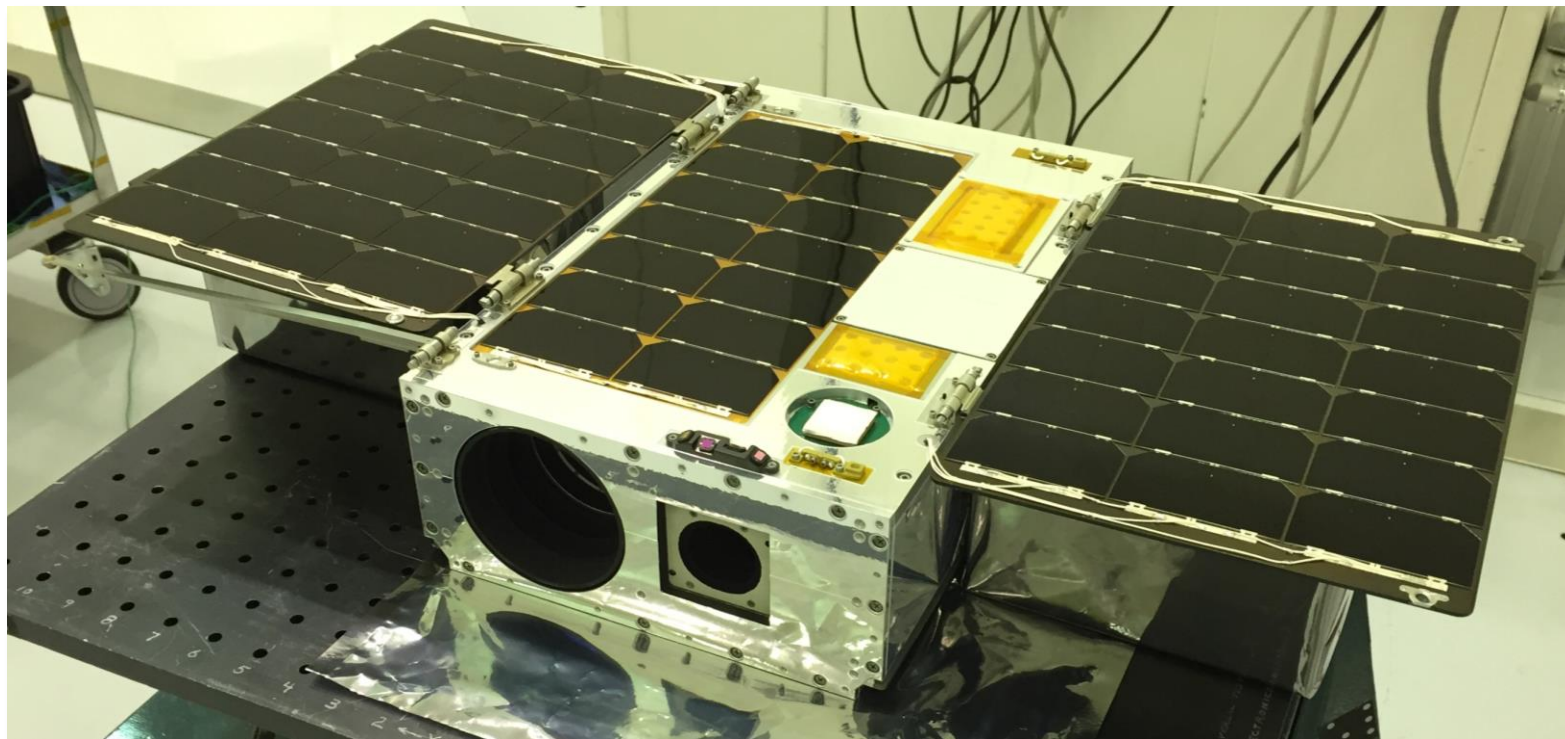
- Quad-core Xeon
- 8GB RAM
- Hard disk



# Our Projects

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY



## ASTERIA

- 6U Cubesat
- Exoplanet imager
- Technology Demonstration

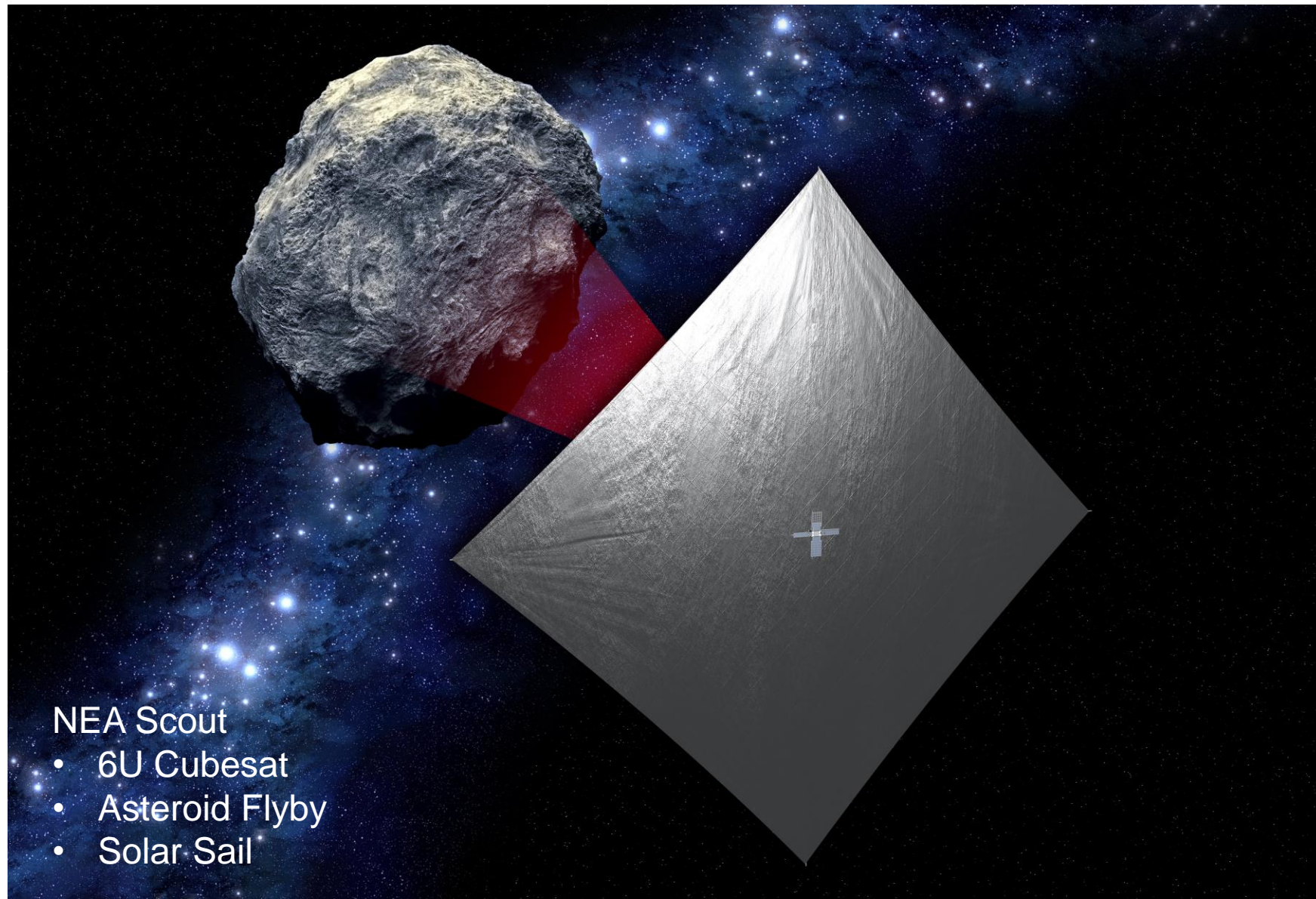




# Our Projects

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY



## NEA Scout

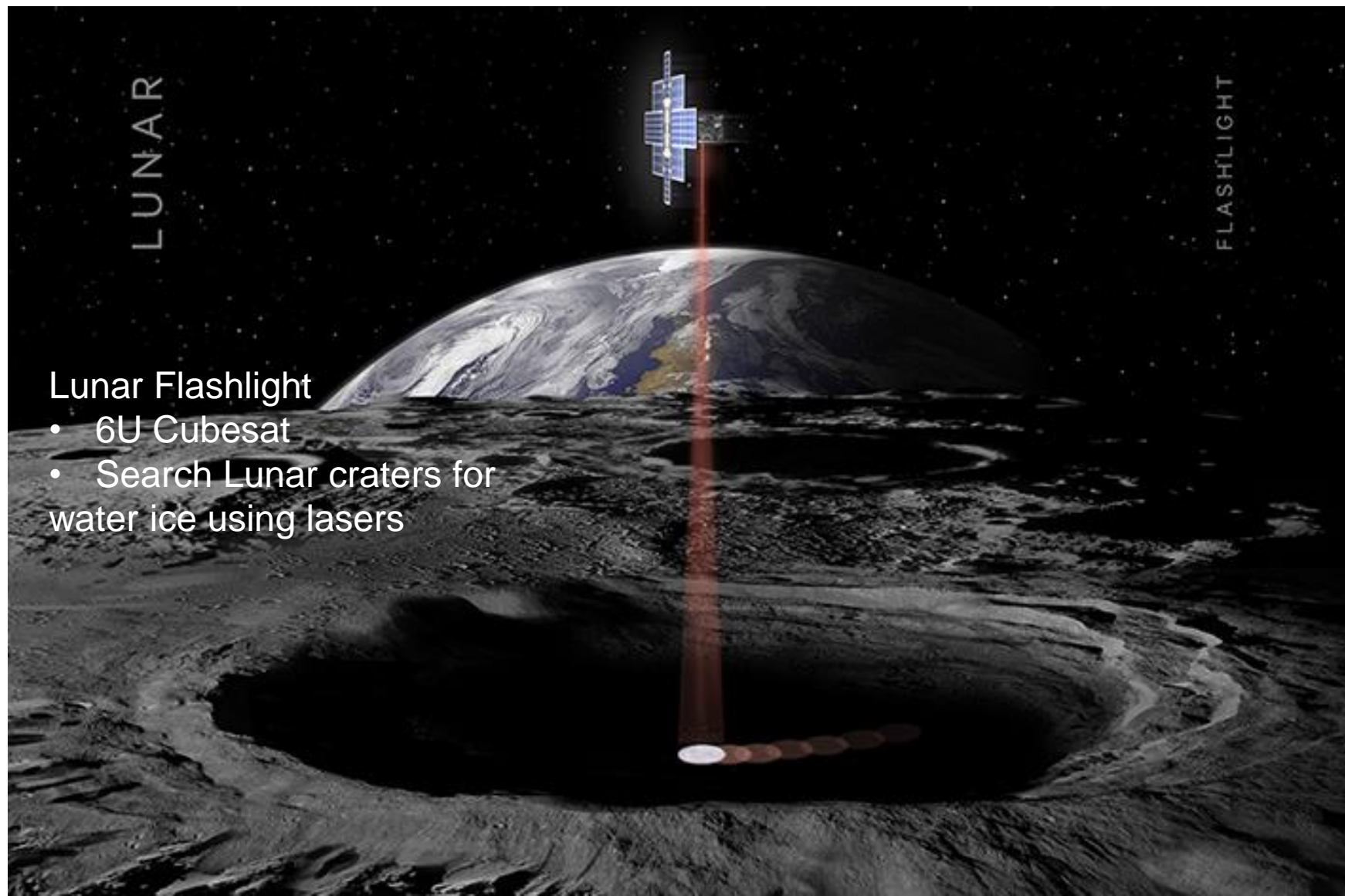
- 6U Cubesat
- Asteroid Flyby
- Solar Sail



# Our Projects

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY



## Lunar Flashlight

- 6U Cubesat
- Search Lunar craters for water ice using lasers



# Our Projects

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

Mars Helicopter  
Technology Development

- Scout for Mars Rover
- Autonomous 2-3 minute daily flights

